

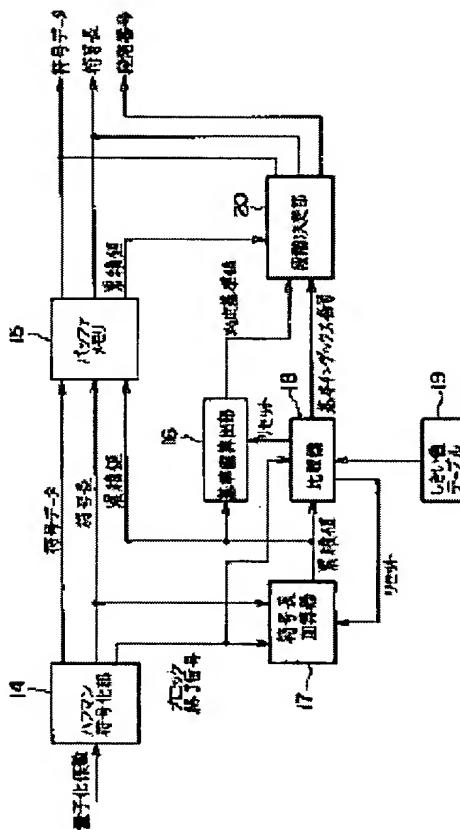
METHOD AND DEVICE FOR PROCESSING PICTURE

Patent number: JP6113142
Publication date: 1994-04-22
Inventor: SAITO KAZUHIRO; others: 01
Applicant: CANON INC
Classification:
 - **international:** H04N1/41; G06F15/66; H03M7/40; H04N7/133
 - **european:**
Application number: JP19930190585 19930730
Priority number(s):

Abstract of JP6113142

PURPOSE: To provide method/device by which a code data amount can be controlled to a target data amount.

CONSTITUTION: In the method/device by which the code data amount can be controlled to the target data amount, a quantized quantizing coefficient is encoded in a Huffman encoding part 14, code data and a code length are stored in a buffer memory 15, the code length is also inputted to a code length adder 17 and the added result is compared with the value of a threshold table 19 in a comparator 18. A stage decision part 20 decides a stage where code data and the code length in the buffer memory 15 are distributed based on an index (compared result) from the comparator 18, a reference value from a reference value calculation part 16 and a code length cumulative value from the buffer memory 15, and outputs it as a stage number.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-113142

(43)公開日 平成6年(1994)4月22日

(51)Int.Cl.⁵
H 04 N 1/41
G 06 F 15/66
H 03 M 7/40
H 04 N 7/133

識別記号 B 9070-5C
330 H 8420-5L
8522-5J
Z

F I

技術表示箇所

審査請求 未請求 請求項の数24(全 19 頁)

(21)出願番号 特願平5-190585
(22)出願日 平成5年(1993)7月30日
(31)優先権主張番号 特願平4-205247
(32)優先日 平4(1992)7月31日
(33)優先権主張国 日本 (JP)

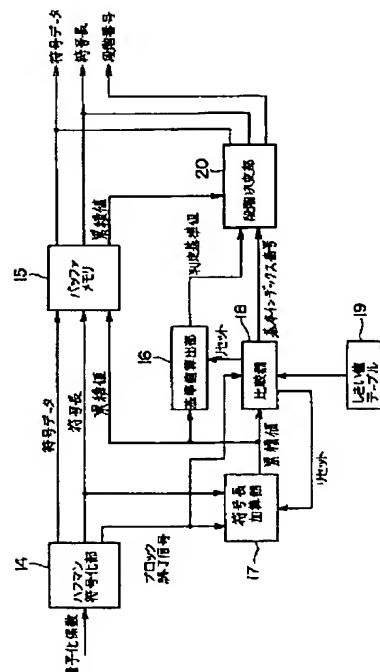
(71)出願人 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(72)発明者 斎藤 和浩
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内
(72)発明者 下村 ゆかり
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内
(74)代理人 弁理士 大塚 康徳 (外1名)

(54)【発明の名称】 画像処理方法及び装置

(57)【要約】

【目的】符号データ量を目標のデータ量に制御できる画像処理方法及び装置を提供する。

【構成】 符号データ量を目標のデータ量に制御できる画像処理方法及び装置。量子化された量子化係数がハフマン符号化部14で符号化され、符号データ及び符号長がバッファメモリ15に格納されると共に符号長は符号長加算器17にも入力され、その加算結果が比較器18でしきい値テーブル19の値と比較される。そして、段階決定部20にて比較器18からのインデックス(比較結果)、基準値算出部16からの基準値及びバッファメモリ15からの符号長累積値に基づいてバッファメモリ15内の符号データと符号長を振り分ける段階を決定し、段階番号として出力する。



1

【特許請求の範囲】

【請求項 1】 画像の周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理装置であって、
 量子化された周波数係数をブロック単位で符号化する符号化手段と、
 該符号化手段から出力される符号データの符号量をモニタするモニタ手段と、
 このモニタ手段によってモニタした符号量に基づいて、前記符号化手段からの符号データに対して段階情報を割り当てる割当手段と、
 この段階情報に応じて、前記符号化手段からの符号データを出力する出力手段とを有することを特徴とする画像処理装置。
 10

【請求項 2】 前記モニタ手段は、各段階に対して、その段階に前もって割り当てられた閾値としての符号数を記憶する手段と、前記符号化手段から出力されてきた符号データの数を前記モニタ手段から入力し、この符号データの数を前記記憶手段から読みだした閾値と比較する手段と、
 20 この比較手段による比較結果に応じて段階情報を決定する手段とを具備することを特徴とする請求項 1 に記載の画像処理装置。

【請求項 3】 前記モニタ手段は、各段階に対して、その段階に前もって割り当てられた閾値としての符号量を記憶する手段と、前記モニタ手段からの前記トータルの符号量と、前記記憶手段から読みだした閾値とを比較する手段と、この比較手段による比較結果に応じて、段階情報を決定する手段とを具備することを特徴とする請求項 1 に記載の画像処理装置。
 30

【請求項 4】 前記出力手段は、複数の符号データを格納するために、前もって所定の単位の大きさのセグメントに分割されたデータ格納手段と、各セグメント毎に、そのセグメントの空き容量を監視する手段と、前記データ格納手段の各セグメントと、各々のセグメントに格納された符号データの属する段階との対応を記憶するテーブル手段とを具備することを特徴とする請求項 40 1 に記載の画像処理装置。

【請求項 5】 前記出力手段は、1つのセグメントに、そのセグメントに割り当てられた段階に属する符号データを新たに記憶するための領域がなくなったと前記監視手段が判断したときに、前記データ格納手段の未使用のセグメントを探し、そのセグメントにこの新たな符号データを格納すると共に、前記テーブル手段のデータを更新する手段を具備するとを特徴とする請求項 4 に記載の画像処理装置。

【請求項 6】 前記出力手段は、未使用のセグメントを 50

2

探すときは、使用中のセグメントに最も近いセグメント割り当てる特徴とする請求項 4 に記載の画像処理装置。

【請求項 7】 前記出力手段は、前記データ格納手段の空きの領域がなくなったときに、より後段の段階に属する符号データを記憶していたセグメントを開放することを特徴とする請求項 4 に記載の画像処理装置。

【請求項 8】 前記出力手段は、1つの符号データと、その符号データに対して前記割当手段によって割り当たされた段階情報を組にして出力することを特徴とする請求項 1 に記載の画像処理装置。

【請求項 9】 前記画像は直交変換によってその周波数成分が抽出され、この周波数成分はHuffman符号によって符号化されることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 10】 前記画像の周波数成分は、既知の周波数シーケンスで入力されて前記符号化手段によって符号化されることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 11】 前記画像の周波数成分は、低周波成分から高周波成分の順序で並んでいることを特徴とする請求項 10 に記載の画像処理装置。

【請求項 12】 画像の周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理方法であって、量子化された周波数係数をブロック単位で符号化する工程と、前記工程で符号化された符号データの符号量をモニタするモニタ工程と、モニタして得た符号量に基づいて、符号データに対して段階情報を割り当てる工程と、この段階情報に応じて符号データを出力する工程とを有することを特徴とする画像処理方法。

【請求項 13】 前記モニタ工程は、各段階に対して、その段階に前もって割り当てられた閾値としての符号数を記憶し、入力されてきた符号の数を、前記トータルの符号量として入力し、この符号の数を読みだした閾値とを比較し、この比較結果に応じて段階情報を決定することを特徴とする請求項 12 に記載の画像処理方法。

【請求項 14】 前記モニタ工程は、各段階に対して、その段階に前もって割り当てられた閾値としての符号量を記憶し、前記トータルの符号量と、読みだした閾値とを比較し、この比較結果に応じて、段階情報を決定することを特徴とする請求項 12 に記載の画像処理方法。

【請求項 15】 前記出力工程は、前もって所定の単位の大きさのセグメントに分割されたデータ格納手段の個々のセグメントの空き容量を監視する工程と1つのセグメントに、そのセグメントに割り当

てられた段階に属する符号データを新たに記憶するための領域がなくなったと判断されたときに、前記データ格納手段の未使用のセグメントを探し、そのセグメントにこの新たな符号データを格納すると共に、前記データ格納手段の各セグメントと、各々のセグメントに格納された符号データの属する段階との対応を記憶するテーブルのデータを更新する工程とを有することを特徴とする請求項12に記載の画像処理方法。

【請求項16】 前記出力工程は、未使用のセグメントを探すときは、使用中のセグメントに最も近いセグメント割り当てることを特徴とする請求項15に記載の画像処理方法。
10

【請求項17】 前記出力工程は、前記データ格納手段の空きの領域がなくなったときに、より後段の段階に属する符号データを記憶していたセグメントを開放することを特徴とする請求項15に記載の画像処理方法。

【請求項18】 前記出力工程は、1つの符号データと、その符号データに対して割り当てられた段階情報を組にして出力することを特徴とする請求項12に記載の画像処理方法。
20

【請求項19】 前記画像は直交変換によってその周波数成分が抽出され、この周波数成分はHufmann符号によって符号化されることを特徴とする請求項12に記載の画像処理方法。

【請求項20】 画像を周波数空間で量子化し、量子化された周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理方法であって、
30

量子化された周波数係数をブロック単位で符号化する符号化工程と、
該符号化工程による符号データの符号長を加算する加算工程と、

該加算工程での加算結果に基づいて段階を判定する段階判定工程と、
該段階判定工程での判定結果に応じて前記符号データを出力する出力工程とを有することを特徴とする画像処理方法。
30

【請求項21】 ブロック順次に入力された入力された画像データに対して異なる圧縮方式で圧縮を行う画像処理装置であって、
40

入力されたブロック順次の画像データを可逆圧縮方式で圧縮する第1の圧縮手段と、
該第1の圧縮手段による圧縮結果に従って、前記ブロック順次の画像データを不可逆圧縮方式で圧縮し、該圧縮データを複数の段階に振り分けて出力する第2の圧縮手段と、

前記第1の圧縮手段で圧縮して得た圧縮データに基づいてそのブロックの特徴を判定する手段と、

前記判定手段によってえた特徴に基づいて前記第1及び第2の圧縮手段での圧縮データのいずれかを選択する手
50

段と、

選択された圧縮データを複数の段階に振り分けて出力する出力手段とを具備する画像処理装置。

【請求項22】 前記判定手段は、文字、図形または自然画像を判定することを特徴とする請求項21に記載の画像処理装置。

【請求項23】 文字、図形または自然画像を対しては前記第1の圧縮手段で圧縮して出力し、また自然画像に対しては前記第2の圧縮手段で圧縮し、該圧縮データを複数の段階に振り分けて出力することを特徴とする請求項22に記載の画像処理装置。

【請求項24】 画像を周波数空間で量子化し、量子化された周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理装置であって、

量子化された周波数係数をブロック単位で符号化する符号化手段と、
該符号化手段による符号データをカウントするカウント手段と、

該カウント手段でのカウント結果に基づいて段階を判定する段階判定手段と、
該段階判定手段での判定結果に応じて前記符号データを出力する出力手段とを有することを特徴とする画像処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、画像の周波数成分をブロック単位で符号化し、この符号化されたデータを複数の段階に振り分けて出力する画像処理方法及び装置に関するものである。

【0002】

【従来の技術】 近年、カラー出力機器の出現に伴い、コンピュータで図形・文字・画像等を扱い、1つの画像を作成するDTPのカラー化が急速に進んでいる。これにより、作成されたカラー画像をコンピュータ等で扱う場合、そのデータ量は膨大なものであるため、そのデータを保存するための大容量メモリや、それを通信するために、多大な通信時間コストが必要となる。これら大容量メモリ、通信時間コストの削減のため、画像データの圧縮という技術が必須となり、様々な圧縮方式が提案されている。

【0003】 カラー多値画像の一般的な圧縮方式としては、JPEG (Joint Photographic Expert Group) によるADCT (Adaptive Discrete Cosine Transform) 圧縮方式があり、ADCT圧縮方式は自然画像を主な対象とした圧縮方式で、不可逆な圧縮方式ではあるが、高圧縮率を期待できるものである。このADCT圧縮方式は次のような圧縮を行う。即ち、3原色 (RGB) 信号をY, U, Vの3成分に変換する。ここで、Yは輝度成分を示し、U, Vは色度成分を示す。輝度成分と色度成分

とを比べた場合、輝度成分の方が色度成分より人間の視覚に敏感であるということを利用して、Y成分はそのままの解像度で圧縮し、U、V成分は場合によっては、サブサンプリングにより解像度を落として圧縮する。

【0004】次に、サブサンプリングにより得られたYUVデータは、各成分について、 8×8 の大きさのブロック毎にDCT変換され、空間周波数成分として抽出される。以下、DCTにより変換された成分を「DCT係数」と称す。そして、これらDCT係数は、輝度成分(Y)と色度成分(U、V)のそれぞれについて設定された 8×8 の大きさの量子化テーブルにより、線形量子化される(除算される)。以下、この量子化された係数を「量子化係数」と称す。最後に、これら量子化係数は、可変長符号化方式であるハフマン符号化方式を用いて符号化される。

【0005】以上がADCT圧縮方式による圧縮の手順である。

【0006】

【発明が解決しようとする課題】しかしながら、上述のADCT圧縮方式の課程で用いられるハフマン符号化方式は可変長符号化方式であるため、その符号化が終了するまで圧縮データ量がわからず、従って目標の圧縮データに制御すること(予め決められた固定長に全体の画像データを圧縮すること)ができないという欠点があった。

【0007】また、このADCT圧縮方式をCG(computer graphics)により作成された文字・図形部に適用した場合、サブサンプリングや量子化の際の誤差が大きくなり、文字・図形部の劣化が大きくなるという欠点もあった。本発明は、上記課題を解決するために成されたもので、符号データ量を目標のデータ量に制御できる画像処理方法及び装置を提供することを目的とする。

【0008】本発明の他の目的は、順次入力してきた符号データのトータルの符号量をモニタし、このトータルの符号量に応じて段階情報をその符号データに割り当てる画像処理装置及び方法を提供することにある。本発明のさらなる目的は、入力されたブロック画像データのと画像の特徴(例えば、文字画像と中間調画像)に応じて、適応的に、可逆符号化若しくは不可逆符号化を施すことにより、適正な符号量に画像を圧縮することのできる画像処理装置及び方法を提案することにある。

【0009】

【発明を解決するための手段】上記課題を達成するための本発明は、画像の周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理装置であって、量子化された周波数係数をブロック単位で符号化する符号化手段と、該符号化手段から出力される符号データの符号量をモニタするモニタ手段と、このモニタ手段によってモニタした符号量に基づいて、前記符号化手段からの符号データに対して段階情報

を割り当てる割当手段と、この段階情報に応じて、前記符号化手段からの符号データを出力する出力手段とを有することを特徴とする。

【0010】また本発明の画像処理方法にかかる発明の構成は、画像の周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理方法であって、量子化された周波数係数をブロック単位で符号化する工程と、前記工程で符号化された符号データの符号量をモニタするモニタ工程と、モニタして得た符号量に基づいて、符号データに対して段階情報を割り当てる工程と、この段階情報に応じて符号データを出力する工程とを有することを特徴とする。

【0011】また他の構成になる本発明の画像処理方法は、画像を周波数空間で量子化し、量子化された周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理方法であって、量子化された周波数係数をブロック単位で符号化する符号化工程と、該符号化工程による符号データの符号長を加算する加算工程と、該加算工程での加算結果に基づいて段階を判定する段階判定工程と、該段階判定工程での判定結果に応じて前記符号データを出力する出力工程とを有することを特徴とする。

【0012】また他の構成になる本発明の画像処理装置は、ブロック順次に入力された入力された画像データに対して異なる圧縮方式で圧縮を行う画像処理装置であって、入力されたブロック順次の画像データを可逆圧縮方式で圧縮する第1の圧縮手段と、該第1の圧縮手段による圧縮結果に従って、前記ブロック順次の画像データを不可逆圧縮方式で圧縮し、該圧縮データを複数の段階に振り分けて出力する第2の圧縮手段と、前記第1の圧縮手段で圧縮して得た圧縮データに基づいてそのブロックの特徴を判定する手段と、前記判定手段によってえた特徴に基づいて前記第1及び第2の圧縮手段での圧縮データのいずれかを選択する手段と、選択された圧縮データを複数の段階に振り分けて出力する出力手段とを具備することを特徴とする。

【0013】また他の構成になる本発明の画像処理方法は、画像を周波数空間で量子化し、量子化された周波数係数をブロック単位で符号化し、該符号データを複数の段階に振り分けて出力する画像処理装置であって、量子化された周波数係数をブロック単位で符号化する符号化手段と、該符号化手段による符号データをカウントするカウント手段と、該カウント手段でのカウント結果に基づいて段階を判定する段階判定手段と、該段階判定手段での判定結果に応じて前記符号データを出力する出力手段とを有することを特徴とする。

【0014】

【実施例】以下、図面を参照して本発明に係る好適な一実施例を詳細に説明する。

〈全体構成〉図1は、本実施例にかかる画像処理装置の

構成を示す概略ブロック図である。図中、CPU11は、本装置全体の制御を行うとともに、各種テーブルの設定を行うためのものである。ROM12には、種々のテーブル等が格納されている。また、RAM13はテーブルの設定等を行うためのワーク領域である。

【0015】以下、画像データを圧縮する場合について述べる。まず、色変換部1は、RGBの入力画像データを以下に示す(1)式で表される 3×3 の線形行列変換によってY, U, V成分に変換する。ここで、Yは輝度成分を、U, Vは色度成分を表す。

10

【0016】

【数1】次に、サブサンプリング部2は、人間の目の感度特性が色度成分(U, V)より、輝度成分(Y)の方に敏感であるということを利用して、色変換部1からのYUV信号に対してサブサンプリングを行う。サブサンプリングは、次の3通りの内の1つの圧縮率でU, Vを圧縮する。その3通りとは、Y : U : V = 4 : 4 : 4 (サブサンプリングを行わない)、Y : U : V = 4 : 2 : 2 (U, Vに対してサブサンプリングを行う)、或いはY : U : V = 4 : 1 : 1 (U, Vに対してサブサンプリングを行う)に変換する。サブサンプリング部2からの出力はそれぞれ 8×8 のブロック単位で出力されるが、個々の信号の出力順序は、Y : U : V = 4 : 4 : 4の場合、Y1, U1, V1, Y2, U2, V2, …の順で、また、Y : U : V = 4 : 2 : 2の場合、Y1, Y2, U1, V1, Y3, Y4, U2, V2…の順で、そして、Y : U : V = 4 : 1 : 1の場合、Y1, Y2, Y3, Y4, U1, V1, Y5, Y6, Y7, Y8, U2, V2, …の順で出力される。

【0017】DCT部3は、これらの出力データを 8×8 ブロック単位でDCT変換してDCT係数を出力する。量子化部4は、そのDCT係数を 8×8 ブロック毎に量子化テーブル8を用いて量子化し、量子化係数を出力する。また、量子化部4は、64個の2次元量子化係数は、図2に示すようにジグザグ順序のスキャンに従った低周波成分から高周波成分へと順に1次元に並べ変えられ、適応ハフマン符号化部5へと送られる。この適応ハフマン符号化部5は、この一次元データを、64個毎に、詳細は後述する図3に示す方法により符号化し、その符号データ、その符号データの長さ(ビット数を表す「符号長」)、その符号データの振り分け段階を表す番号(「段階番号」と呼ぶ)をそれぞれ出力する。尚、実施例では、設けられた段階の数を一例として4つ設定している。

【0018】セグメントコントローラ6は、上述の「符号データ」、「符号長」、「段階番号」に基づいてその「段階番号」で示された段階における圧縮データを圧縮メモリ7に書き込む。この圧縮メモリはセグメントに区切られており(図6を参照)、各段階毎に、その段階に対応するセグメント内に圧縮データが書き込まれる。ま

50

た、このセグメント単位に振り分けた情報は同時にセグメント情報テーブル10(図7を参照)にも書き込まれる。情報テーブル10に書き込まれたデータは伸長の際に用いられる。このセグメントコントローラ6によって段階別に振り分けられたデータを格納するセグメントを制御することにより、固定長圧縮を実現することができる。

【0019】この「固定長圧縮」を実現することのできる理由を簡単に説明する。前述したように、ADCT変換された量子化係数は、順に、直流成分→低周波交流成分→高周波交流成分と並んでいる。周波数のより低い成分の符号データに「より低位の段階番号」を割り当て、周波数のより高い成分の符号データに「より高位の段階番号」を割り当てるようになると、「より高位の段階番号」を割り当てられた符号をより多く出力若しくは格納するということは、原画像の情報をより精度良く保存することを可能にする。即ち、「より高位の段階番号」の符号データを出力若しくは格納することは、より精度高く「圧縮」するということを意味する。そこで、各段階に振り分けられるデータは、バッファメモリの容量や対象画像の圧縮データの量の大きさに従って、1, 2段階のみで「圧縮を行う」(出力若しくは記憶する)、あるいは1, 2, 3段階のみで「圧縮を行う」(出力若しくは記憶する)、或いは1, 2, 3, 4の全ての段階で「圧縮を行う」(出力若しくは記憶する)というように、圧縮を行うべき段階を選択することにより、目標のデータ量に制御することができる。例えば、第1段階で「出力」できるデータ量が2.5MB、第2段階で「出力」できるデータ量が1.5MB、第3段階で出力できるデータ量が0.8MB、第4段階で「出力」できるデータ量が0.5MBであって、目標のデータ量が5.0MBであれば、第1, 2, 3段階の3つのデータを用いることにより、全圧縮データ量は4.8MBとなり、圧縮を目標のデータ量以内に制御することができる。このように、全符号データの段階別の振り分け方により固定長圧縮の精度が影響される。上述のセグメントコントローラ6、圧縮メモリ7、セグメント情報テーブル10に関しては、後で詳細に説明する。

〈第1実施例…圧縮動作〉次に、図3を参照して上述の適応ハフマン符号化部5の概略を説明し、その後、各ブロックの詳細を説明する。

【0020】このハフマン符号化部5の概略は、画像全体の目標の「全体目標ビットレート数」(「ビットレート」とは、圧縮率を示す単位で、1画素当たりのビット数 [bits/pixel] である)と第1段階の目標ビットレート数が設定されたならば、第1段階における符号化を行なった後に、そのブロックにおける残りの目標ビットレート数(=「全体目標ビットレート数」-「第1段階目標ビットレート数」)をおよそ3等分し、等分したビットレートをそれぞれ第2, 第3、第4段階に振り分ける

ように、各段階において、「符号データ」、「符号長」、「段階番号」を出力するというものである。例えば、「全体目標ビットレート」を 2.4 bit/pixel とし、 8×8 ブロックの画像に対して、「第1段階目標ビットレート数」を 1.5 bit/pixel とすると、第2, 第3、第4段階における「目標ビットレート数」は夫々 0.3 となる。

【0021】適応ハフマン符号化部5の詳細は図3に示される。同図に示されるように、適応ハフマン符号化部5は、ハフマン符号化部14とバッファメモリ15と基準値算出部16と符号長加算器17と比較器18と段階決定部20とを具備する。符号化部5は、前述したように、符号データと符号長と段階番号とを出力する。符号化部14が符号化した符号データの各々には、どの「段階番号」を割り当てるかは、段階決定部20が決定する。符号長加算器17、比較器18、基準値算出部16は、各符号データの符号長値を累積し、その累積値と所定の閾値との比較に基づいて、その閾値を越えようとするときの「インデックス番号」と、別途決定された「基準値」とを生成する。決定部20は、この「インデックス番号」と「基準値」とに基づいて「段階」を決定する。バッファメモリ15は、段階の判定に時間を要するために、決定部20から出力される「段階番号」と、符号化部14で生成された「符号データ」と「符号長」とを同期させて出力するために必要となる。以下、更に図3のシステムについて説明する。

【0022】図3において、ハフマン符号化部14は、1次元に並べられた量子化係数を64個毎(8×8 ブロックの各Y, U, V成分毎)にハフマン符号化し、符号データと符号長とをバッファメモリ15に、そして、ブロックの区切りを示すブロック終了信号を符号長加算器17に出力する。バッファメモリ15は、これらのデータを図4に示すような形で格納する。

【0023】図4に示したバッファメモリ15は、「インデックス」フィールドと、ハフマン符号を格納する「符号データ」フィールドと、その符号の長さを格納する「符号長」フィールドと、符号長の累積加算値を格納する「符号長累積値」フィールドとからなる。「インデックス」は、1つのブロックにおける出力された符号データの番号を示す。また、符号長累積値は符号長加算器17から得られる。

【0024】図4の例では、最初のブロックについての「インデックス」フィールドには、 $0 \sim 18$ の値が格納されているので、そのブロックは19個のハフマン符号に符号化されたことが分かる。また、符号化部14に入力される量子化係数は、DCT部3においてDCT変換されているので、最初の符号データは直流(DC)成分であり、以降の符号データは交流(AC)成分となる。図4の例では、第1ブロックのインデックス0はDC成分を、インデックス1～18はAC成分を示している。

【0025】符号長加算器17は、ブロック終了信号に同期して各ブロック毎にAC成分の符号データの符号長を累積加算していく、その累積値を比較器18に出力する。比較器18は、この累積値との比較対象となる閾値を、Y, U, V毎に、閾値テーブル19に用意している。図5は、その閾値テーブルの一例を示す。この例では、各閾値は、Y成分については“40”、U成分については“28”、V成分については“28”を設定している。比較器18は加算器17にリセット信号を出力する。即ち、比較器18は、累積値を閾値と比較する。累積値がしきい値を越えることとなる場合には、その越えることの原因となる符号データの符号長の加算は行なわずに、前の符号データのインデックス番号(以下、このインデックス番号便宜上「基本インデックス番号」と呼ぶ)を段階決定部20に出力するとともに、符号長加算器17をリセットする。加算器17は、基本インデックス番号の次の符号データから、再び符号長の加算を始める。加算器17は、ブロック終了信号によりブロックが終了したことを検出したならば、その時点での累積値(即ち、リセットされた後のブロックが終了するまでの符号長の累積値)を基準値算出部16に送る。

【0026】基準値算出部16では、この最終の累積値から複数通りの基準値を算出する。この基準値は決定部20に出力され、決定部20は、この基準値と比較器18からの基本インデックス番号とにに基づいて段階を判定する。以下、第1の実施例における適応ハフマン符号化部5の動作を図4及び図5を参照して具体的に説明する。

【0027】図5は、第1段階の目標ビットレートを 1.5 [bits/pixel] として実現しようとするときの、Y, U, V成分夫々に対する閾値の例であり、それぞれ40, 28, 28 [bits]としている。図4は、Y成分のある 8×8 ブロックについての、符号データ、符号長、符号長累積値を示す。図示するように、インデックス番号0はDC成分であるために、加算器17は、そのDC成分の符号長は加算の対象とはせず、インデックス=1(AC成分)の符号データの符号長(=18)から累積値フィールドに書き込む。インデックス番号2のデータの符号長は7なので、累積値は $18+7=25$ となる。同様に、インデックス番号3の符号長累積値は39となる。次に、インデックス番号4の符号長が“21”なので、符号長累積値は“60”となるところであるが、この“60”は、比較器18によりY成分のしきい値(40 bits)と比較され、 $60 > 40$ であるので、その1つ前のインデックス番号“3”が「基本インデックス番号」として段階決定部20に送られる。そして、符号長加算器17は比較器18によってリセットされ、またインデックス番号4の符号長“21”がそのまま符号長累積値に書き込まれる。インデックス番号=5では、符号長が“9”であるので符号長累積値は、 $21+9=30$ となる。

30となる。

【0028】以下同様に、各符号データの符号長が加算され、その結果は、そのブロックが終了するまで符号長累積値フィールドに書き込まれる。インデックス=18において、符号長累積値の最終値“95”は、基準値算出部16に転送される。基準値算出部16は、残りの段階数が3つあるので、95を3で除算し、その商の“30”とその2倍の“60”とが「判定基準値」として段階決定部20に送られる。

【0029】段階決定部20は、比較器18から得たインデックス番号（累積値が閾値を越えようとするときのインデックス番号）と、基準値算出部16から得た「判定基準値」とに基づいて、各符号データと符号長データに対して付与すべき「段階」を決定する。決定部20の動作を具体的に説明する。前述したように、バッファメモリ15には、図4に示したごときデータが格納されている。バッファメモリ15からは、「符号データ」と「符号長」データと「符号長累積値」とが同期して出力される。これら3つのデータのうちの「符号長累積値」は、決定部20に入力される。一方、決定部20には、比較器18からの「基本インデックス番号」が入力されている。そこで、決定部20は、「基本インデックス番号」以前のインデックスの符号データと符号長については、段階=1を与える。図4の例では、「基本インデックス番号」=3であるので、インデックス番号=0~3までの符号データと符号長データとは、段階=1として扱う。決定部20は、「基本インデックス番号」以降のインデックス番号のデータに対しては、前述の複数個の判定基準値に基づいて段階番号を決定して付与する。即ち、決定部20は「判定基準値」として“30”，“60”を基準値算出部16から入力している。そこで、決定部20は、バッファメモリから読みだした「符号長累積値」と「判定基準値」（=30）とを比較し、「符号長累積値」が30以下であることを示す「符号データ」と「符号長」データに対しては、段階番号=2を付与し、31以上の60以下の場合には段階番号=3を付与し、また61以上の場合には段階番号=4を出力する。ここで、各段階の区切りには、符号データ間に、必ずその段階の区切りを示すEOS(End of Stage)コードが挿入される。

【0030】以上のようにして、 8×8 (=64個の係数)のブロック毎に、複数の段階に振り分けられて圧縮を行うことができた。上記例では、第1段階のデータについては、およそ1.5 bits/pixelのビットレートが割り振られ、残りのデータについてはビットレートが3等分されて圧縮された。上述の図4、図5の例では、Y成分については符号長累積値が“40”を越えたが、40を越えないようなブロックの画像データの圧縮の場合には、途中でEOB(End of Block)コードが挿入され、その 8×8 ブロックの符号データは、全て第1段階のセ

グメントに書き込まれる。

【0031】こうして得られた「符号データ」と「符号長」と「段階番号」とはセグメントコントローラ6（図1）に送られ、「符号データ」と「符号長」とは「段階番号」にしたがってセグメント単位に圧縮メモリ7に格納される。

〈第1実施例…伸張動作〉次に、前述したプロセスにより圧縮されたデータを伸長するプロセスについて以下に述べる。

【0032】図1のシステムは、圧縮にも伸張にも動作可能である。即ち、図1のシステムは、データの伸張を行う場合には、データの流れを逆にするだけで伸張器として動作する。この場合、DCT部3は逆DCT部3'となり、量子化部4は逆量子化部4'となり、そして、適応ハフマン符号化部5は適応ハフマン復号化部5'となる。また、量子化テーブル8とハフマンテーブル9は、それぞれ逆量子化テーブル、ハフマン復号化テーブルとなり、データの流れは、圧縮の時と逆の流れとなる。

【0033】前述したように、1つの 8×8 ブロックを構成する符号データが複数の段階に分かれて圧縮メモリ7に格納されている。そこで、適応ハフマン復号化部5'は、まず、セグメントコントローラ6に対して段階番号=1の符号データを要求する。セグメントコントローラ6は、セグメント情報テーブル10の内容を照らし合わせ、圧縮メモリ7から第1段階の符号データを読み取って適応ハフマン復号化部5'に転送する。適応ハフマン復号化部5'では、得られた符号データを次々と復号化し、その結果を逆量子化部4'に転送し、EOS(End of Stage)を検出するまで上述の処理を繰り返す。

【0034】復号化部5'は、EOSを検出したならば、次の段階番号2の符号データをコントローラ6に対して要求する。セグメントコントローラ6は、上述の如く、セグメント情報テーブル10の内容を照らし合わせ、圧縮メモリ7から第2段階の符号データを読み取り、適応ハフマン復号化部5'に転送する。適応ハフマン復号化部5'では、圧縮と同様に、EOB(End of Block)コードを検出するまで復号化を行う。以下同様に第3、第4段階の復号化を行い、1つの 8×8 ブロックの復号化が終了する。この場合において、途中でEOBを検出すると、次の段階の符号データの要求は行わず、次の 8×8 ブロックの復号化を始める。また、画像の圧縮データ量や圧縮メモリの容量により、最終段階まで圧縮メモリに格納できなかった場合は、途中の段階までのデータを用いて伸長を行う。

【0035】このようにして得られた量子化係数は、逆量子化部4'にて逆量子化テーブル8'を用いて逆量子化され、逆DCT部3'に送られる。逆DCT部3'では、得られたDCT係数を逆DCT変換し、Y' U'

V' データを得る。サブサンプリング部2では、サブサンプリングの比 (Y : U : V = 4 : 4 : 4, 4 : 2 : 2, 4 : 1 : 1) に応じて拡大操作が実行される。色変換部1では、以下に示す(2)式に従って、逆変換がなされ、元の画像が復元される。

【0036】

【数2】〈第1実施例…セグメンテーション〉ここで、画像データの圧縮処理及び伸長処理における、多段階に分割された圧縮データを、複数のセグメントから構成される圧縮メモリ7へ格納し、選択したセグメントのセグメント情報をセグメント情報テーブル10へ格納する処理について以下に詳細に説明する。

【0037】圧縮メモリ7は、図6に示すように、S-1からS-Nまで区切られた複数のセグメント（例えば、1セグメント=100 kB）からなる。図7は、セグメント情報テーブル10の構成を示す。図7のセグメント情報テーブル10の2~11…の値はテーブルの列の番号を示し、左から右へ進むに従い、画像の先頭から終端に進むことを示す。また、同テーブル10の1列目の4つのデータは、夫々、対応する段階が有効であるか／無効であるかを示す。この図7の例では、段階1~3までが有効で、段階4が無効となっている。また、2列目から11列目の各列の4つのデータは、各段階の符号データが書き込まれた圧縮メモリ7内の選択されたセグメント番号 (S-1~S-N) を示す。また、ENDは、各ステージにおける符号データが終了したことを示す。

【0038】次に、段階毎に符号データを圧縮メモリ7に書き込む場合の処理手順、即ち、各段階における符号データをどのセグメントに割り当てるかについて説明する。この割当は：

①：異なる画素についての、同一段階の符号データは、…空き領域があるかぎりは同じセグメントに記憶する。

②：ある段階の符号データを格納するためのセグメント…が満杯になったときは、他の段階によっても使われていないセグメントの最初のセグメントを使用する。このような割当は、圧縮メモリ7を虫食い状にしないで有効に使うことを可能にする。この割当について具体的に説明する。

【0039】図8は、上述の圧縮処理における圧縮データの格納処理を説明するフローチャートである。各段階に振り分けられる圧縮データは、段階の振り分けを行う圧縮方式や原画像データの特性にもより、各段階における符号量も異なってくるが、ここでは、段階1~4を比較した場合、段階1の方が4より出力される符号量が多くなる例を用いる。

【0040】以上を踏まえて、以下、図7の例を用いて図8のフローチャートの制御手順を説明する。図8のステップS1において、最初のブロックの画像データについて1~4の各段階で出力された符号データは、それぞ

れセグメントS-1, S-2, S-3, S-4に書き込まれる。こうして、処理がスタートする。次以降のブロックについて発生する4つの段階の各々についての符号データも、夫々、セグメントS-1, S-2, S-3, S-4に書き込まれるであろう。この書き込み動作はステップS2で、いずれかのセグメントが満杯になることが検出されるまで継続する。この方式の圧縮方法によれば、段階=1の圧縮は必ず適用されるので、段階=1の符号データは他の段階に比べてより大量に発生する。従って、段階=1の符号データが格納されるセグメントS-1はセグメントS-2~S-4よりも早く一杯になることが予想される。そこで、ステップS2でセグメントS-1が満たされたことが検出されると、ステップS3で未使用で最初のセグメントS-5に段階=1の符号データの書き込みが始まる。

【0041】段階=2, 3のための符号データを記憶するセグメントS-2, S-3は、そのいずれかの空き領域が満たされると、満杯になったほうの段階のために、新たなセグメントが割り当てられる。図7の例では、段階2のためのセグメントS-2が段階3のためのS-3よりも先に満杯になったので、段階2のためのセグメントが領域S-6に割り当てられた。次に、段階=3のデータを格納するS-3が満杯になったので、空いているS-7が割り当てられる。

【0042】さらに、段階=4のための符号データを格納するセグメントS-4が一杯になる前に、段階1の符号データを書き込むためのセグメントS-5への書き込みを終了してしまったために、空いているセグメントS-8は、段階=4に割り当てられず、段階=1に割り当てられる。これは、段階1で発生する符号量が、段階4よりも多くなるために起る。以下同様に、各段階において書き込んでいるセグメントが満たされたならば、空いているセグメントを選択して、そこに符号データを書き込むようとする（ステップS2乃至S4）。

【0043】実施例では、原画像をいくつかの段階に分けて符号化する方法を用いている。そこで、各段階の符号化データ量が目標の圧縮メモリ7の容量より小さい場合は、即ち、セグメントS-Nが割当てられるまえに全画像の圧縮を終了する様な場合は。メモリ容量の問題は起きない。しかしながら、全画像の圧縮を終了する前に、最後のセグメントS-Nが割当てられたような場合（ステップS4でYES）は、より重要な符号データ、即ち番号がより若い段階（図7の例では段階=1）の符号データのために、より大きな番号の段階（図7の例では段階=4）の符号データに使われていたセグメントを開放すべきである。図8のステップS5はその開放手続きのためにある。

【0044】即ち、図7の例では、最終セグメントS-Nは段階=2の符号データを格納するために使われた。即ち、この時点では、圧縮メモリ7には空き領域はな

い。そこで、より重要度の低い段階=4の符号データを格納するセグメント（図7の例では、S-4, S-1 1, S-1 5）を開放して、これらのセグメントを段階=1や段階=2のための符号データの格納領域とする。図7の例では、セグメントS-4, S-1 1, S-1 5には、夫々、段階=1, =2, =1の符号データが格納される。そして、セグメント情報テーブル10の段階=4を示す第4行目は、無効データしか格納されていないので、第1列目の対応ビットを0とする。このようにして、圧縮メモリ7が不足した場合には、より重要度の低い段階4を無効（例えば“0”）とし、段階4で用いられていたセグメントS-4を段階=1に用い、セグメントS-1 1を段階=2に用いる。段階=2は、セグメントS-1 1で符号化が終了したので、以降、ENDマークを付ける。また段階=1では、段階=4で用いられていたセグメントS-1 5が割り当てられ、以後、符号化が終了したので、ENDマークが付けられる。

【0045】圧縮メモリ7に格納された圧縮データの伸長に関しては、伸長の際に有効段階1～3のみを用いて画像データを復号する。

〈第2実施例〉次に、第2の実施例におけるインデックスに基づく適応ハフマン符号化部5の構成及び動作を図9に示すブロック図を参照して以下に説明する。

【0046】前述の第1実施例は、符号長を累積し、累積された符号長が所定の値になるかいかを判断することにより、段階番号を決定するというものであった。第2実施例は、段階=1の決定は第1実施例の手法と同じであるが、段階=2以降の決定はインデックス番号に基づいて行うというものである。図9において、ハフマン符号化部21は、1次元に並べられた量子化係数を64個毎に（ 8×8 ブロックの各Y, U, V成分毎に）ハフマン符号化し、符号データと符号長とをバッファメモリ22に出力し、符号長データとブロックの区切りを示すブロック終了信号とを符号長加算器25に出力する。バッファメモリ22では、符号データと符号長データを図10に示すような形式で格納する。図10において、「インデックス」は、第1実施例の「インデクス」番号と同じであり、1つのブロックにおける出力された符号データの番号を示し、符号データのうち有効ビット数は、第1実施例と同じように「符号長」で指定される。

【0047】符号長加算器25は、ブロック終了信号に同期しながら、各ブロックごとに符号データ（AC成分のみ）の符号長を加算累積していく、その累積値を比較器26に出力する。比較器26は、Y, U, Vごとに用意されたしきい値と前記累積値とを比較する。この閾値は、第1実施例と実質的に同じ構成の閾値テーブル27から供給される。加算値がしきい値を越えると比較器26が判断した場合には、その越える前のインデックスを「基本インデックス」としてカウンター23と段階決定部24に出力する。かくして、第2実施例の「基本イン

デクス」は第1実施例の「基本インデクス」と同じ意味を有することとなる。

【0048】閾値テーブル27が図5に示したような閾値を有し、符号データが図10の様な例である場合には、インデックス=4のときに累積値が閾値（=40）を越えることとなるので「基本インデックス」は“3”となる。第2実施例では、「基本インデックス」番号以前のインデックスの符号データは、即ち、図10の例でインデックス0から3までの符号データは段階=1として決定され、段階=1として圧縮メモリ7に書き込まれることとなる。

【0049】また、カウンター23は、入力された1つのブロックの全ての符号データの数をカウントする。即ち、カウンタ23は、1ブロックの最初のAC成分からインデックスをカウントする。ブロック終了信号により1つのブロックが終了したことをカウンタ23が検出したならば、その最終のインデックス番号（「最終インデックス」と呼ぶ）と、比較器26から得られた「基本インデックス」とから複数の「基準インデックス」を算出し、それらを段階決定部24に出力する。

【0050】図10に示す例では、「基本インデックス」番号は“3”となり、また、最終インデックス番号は“18”であるので、段階=2, =3, …が割り当てられるべきデータはインデックス“4”から“18”であると判定できる。そこで、インデックス4～18の全ての符号データを3等分し、各々の分割された符号データグループに、段階=2, =3, …を割り当てる。具体的には、 $(18 - 3) / 3 = 5$ であるから、各グループ内の符号数は“5”となり、「基準インデックス」として、“8”（=3+5）と“13”（=8+5）を夫々出力する。また、ハフマン符号化部21からのブロック終了信号は符号長加算器25と比較器26のリセットも行う。

【0051】段階決定部24は、バッファメモリ22からの符号長信号からインデックスをカウントする。これにより、決定部24は、バッファメモリから出力されつつある符号データのインデックス番号をモニタすることができる。一方、決定部24は、カウンタ23から「基準インデックス」番号（図10の例では、8, 13）を、比較器26から「基本インデックス」番号（図10の例では3）を入力する。決定部24は、図10の例では、基本インデックスは3であり、基準インデックスは8, 13であるので、図10に示すインデックス3までのデータに対しては段階番号=1を、4から8までのデータに対しては段階番号=2を、9から13までのデータに対しては段階番号=3を、それ以降このブロックが終了するまでは、段階番号=4を出力する。その際、段階番号の変わり目には、EOS(end of Stage)コードを挿入する。このコードは、伸長する際に必要となるものである。

【0052】また、この符号化方法による符号データを伸長するための適応ハフマン復号化部は、前述した第1の実施例と同様な方法で実現できる。

〈第3の実施例〉次に、第3実施例におけるインデックスに基づく適応ハフマン符号化部5の構成及び動作を図11に示すプロック図を参照して以下に説明する。この第3実施例も、入力される符号データの数をモニタし、まず、ハフマン符号化部31では、1次元に並べられた量子化係数を64個毎に(8×8ブロックの各Y, U, V成分毎に)ハフマン符号化し、符号データと符号長を10出力する。

【0053】次に、カウンター32は、符号長信号をカウントする。このカウントにより、カウンタ32は、入力される符号データの数、即ち、インデックス番号を知ることができる。カウンタ32は、その結果、即ち、インデックスを比較器33に転送する。この比較器33は、図12に示すようなインデックステーブル35の値と比較する。図12に示されたインデックステーブルは、各段階に割り当てられるべき符号データの数を、インデックス番号として記憶したものである。即ち、図12において、Y成分については、インデックス番号1～3までが段階=1に、インデックス番号4～8までが段階=2に、インデックス番号9～15までが段階=3に、インデックス番号15以上が段階=4に、夫々割り当てられる。比較器33は、これらのインデックステーブルからの夫々の値を順に、カウンタ32の出力と比較する。そして、カウンタ32からの出力値がテーブルの値を越える度に、カウンター34にカウントアップ信号を送る。カウンター34の出力は段階番号を示し、その初期値は1となっており、カウントアップ信号が入力されると、出力値をインクリメントする。即ち、段階番号が1つ上がることになる。

【0054】この第3実施例の圧縮手法によって圧縮されたデータを図1の圧縮／伸張装置を用いて伸張する場合には、適応ハフマン復号化部5'は、セグメントコントローラ6に段階番号1の符号データを要求する。セグメントコントローラ6は、セグメント情報テーブル10の内容を照らし合わせて、圧縮メモリ7から第1段階(番号=1)の符号データを読み取り、適応ハフマン復号化部5'へと転送する。適応ハフマン復号化部5'は、得られた符号データを復号化し、インデックステーブル35を参照し、次に要求する段階番号を出力する。以下同様に、段階番号の要求と、得られた符号データの復号化を繰り返し、復号化された量子化係数が64個(8×8ブロックの要素数)となるまで行う。64個復号化されたならば、次の8×8ブロックの復号化を行う。

【0055】前述した第1の実施例(図3)や第2の実施例(図9)では、符号長に基づき、ビットレートの段階別振り分けを行っているが、それに比べ、この第3実50

施例の方法(図11)を用いた場合、その精度は落ちるが、アルゴリズムが簡単なためハードウェア規模が小さくなり、低コストで実現できる。また、1つの8×8ブロックの段階の区切りにEOSコードを挿入する必要がなく、その分符号データ量が減る。

〈第1～第3実施例の変形〉上述した実施例では、圧縮データを振り分ける段階数を4段階として説明したが、これは、2, 3、或いは5, 6など幾つでも良い。

【0056】また、画像全体の目標ビットレートを2.4 [bits/pixel]としたが、この値に限らず幾つでも良い。更に、第1段階の目標ビットレートを1.5 [bits/pixel]とし、それに基づき、図5のしきい値テーブルの値を決定したが、この値に限らず幾つでも良い。また、第2段階以降の符号データをおよそ3等分するように振り分けたが、それだけに限らず、第2, 3, 4段階の比を、例えば3:2:1, 5:3:1など、どのように振り分けても良い。尚、第3の実施例において、段階振り分けの基準となるインデックステーブルとして図9に示すものを用いたが、その値はこれに限るものではない。

〈第4実施例〉次に、図面を参照して本発明に係る第4の実施例を詳細に説明する。

【0057】この第4実施例では、カラー画像に含まれる文字、図形に対しては可逆圧縮方式で圧縮を行い、また自然画像に対しては前述した第1～第3実施例における不可逆圧縮方式で圧縮し、その圧縮データを複数の段階に振り分けて記憶するというものである。図13は、第4実施例にかかる画像圧縮装置の構成を示す概略プロック図である。図において、41はラスター・ブロック変換部であり、ラスター順次で入力される1画素当たり24ビットの画素データをブロック(8×8画素)単位で出力する。42はブロック遅延部であり、1ブロック分の画像データを一時保存することにより、画像データを遅延させる。43は画像データを不可逆圧縮方式で圧縮する不可逆圧縮部であり、ブロック遅延部42からのブロック順次データを圧縮し、圧縮された符号データを複数の段階に分割して出力する。44はセグメントコントローラであり、前述した実施例と同様に多段階に分割された符号データを複数のセグメントに区切られた第1記憶部45に段階別に格納することができる。この格納手段により、固定長圧縮を実現でき、段階別に選択したセグメント番号はセグメント情報テーブル46に格納される。

【0058】可逆圧縮部47は、ブロック順次で入力された画像データをブロック単位に可逆方式で圧縮し、その結果を圧縮データ保持部48に格納すると共にその圧縮データ量をカウントする。そして、カウント結果をDとし、そのブロックに許容されている限界値をLとしたときに、 $L \geq D$ ならば、そのブロックを領域0とし、 $L < D$ ならば、そのブロックを領域1とする。そして、そ

の結果（「判定信号」）を第2記憶部49に格納する。従って、第2記憶部49は、1ブロックを領域0か領域1かを区別する1ビットで表したところのブロックマップとなる。即ち、領域1と判定されたブロックは、その符号量Dが限界値Lを越えたブロックと判断されることとなる。

【0059】この第4実施例の圧縮システムは、可逆符号化方式で圧縮された圧縮符号量が限界値Lを越えないようなブロック（領域0）については、その可逆符号化方式での符号を採用し、Lを越えたようなブロック（領域1）については、非可逆符号化を施すようとしている。図13において、不可逆圧縮部43は、1ブロック遅延されたデータを符号化しているので、セグメントコントローラ44の入力には、1つのブロックの画像データの可逆圧縮された符号データと不可逆圧縮された符号データとが同期して揃っている。また、可逆圧縮部47は、前述の判定信号を圧縮データ保持部48と不可逆圧縮部43とに送っている。即ち、可逆圧縮部47は、領域0と判断したブロックに関しては、その圧縮データが保存されている圧縮データ保持部48のデータをセグメントコントローラ44に転送せしめる。また、領域1と判断されたブロックに関しては、そのブロックの圧縮データ保持部48に保存されている圧縮データを無効とせしめ、そのブロックを不可逆圧縮方式で圧縮するために、その命令信号を不可逆圧縮部43に転送して、不可逆圧縮部43にて圧縮せしめる。尚、不可逆圧縮部43の構成及び動作については、前述した実施例と同様であり、ここでの説明は省略する。

【0060】ここで、図13のシステムにカラーDTPにより作成されたところの文字・図形・自然画像等を含むカラー画像が入力された場合を考える。文字・図形等のCG画像は可逆圧縮方式で圧縮した場合は圧縮効率が高いので、D < Lとなり、復元性に優れる可逆圧縮方式で圧縮した方が好ましい。一方、自然画像は画素値のバラツキが大きいので、可逆圧縮を行ってもデータ量が小さくなることは期待できない。従って、自然画像に対しては圧縮効率の高い不可逆圧縮方式で圧縮することが望ましい。

【0061】第4実施例のプロセスによれば、カラーDTPにより作成された文字・図形・自然画像等を含むカラー画像を、文字・図形等のCG画像は可逆圧縮方式で圧縮し、また自然画像に対しては不可逆圧縮方式で圧縮することができる。次に、圧縮されたデータを伸長するプロセスについて図14を参照して以下に説明する。

【0062】図14は、第4実施例での画像伸長装置の構成を示す概略ブロック図である。図において、図13と同様なものには同一の符号を付している。この伸張装置は、図13のシステムにより第1記憶部45と第2記憶部49とに格納されたデータに基づいて伸張を行う。セグメントコントローラ44は、不可逆伸長部51、或

いは可逆伸長部52の要求する符号データと符号長に応じて、セグメント情報テーブル46より複数のセグメントで区切られた第1記憶部45のアドレスを生成し、符号データを不可逆伸長部51、或いは可逆伸長部52に転送する。

【0063】不可逆伸長部51は、図13の不可逆圧縮部43によって圧縮されたデータを伸長するためのもので、自ら要求する符号データの符号長及び段階番号を生成する。そして、得られた符号データを伸長して、ブロック単位でスイッチ部53に転送する。可逆伸長部52は、図13の可逆圧縮部47によって圧縮されたデータを伸長するためのもので、自ら要求する符号データの符号長及び段階番号を生成する。そして、得られた符号データを伸長して、ブロック単位でスイッチ部53に転送する。

【0064】スイッチ部53では、第2記憶部49に格納されているビット情報に基づいて入力データを選択して出力する。例えば、ビット情報が“0”的場合、可逆伸長部52からのデータを選択し、“1”的場合、不可逆伸長部51からのデータを選択する。そして、ブロッククラスター変換部54において、スイッチ部53から出力されたブロック順次のデータがラスタ順次に変換され、伸長プロセスが終了する。

【0065】この第4実施例によれば、可逆圧縮方式と不可逆圧縮方式を併用することにより、CGにより作成された文字・図形部の劣化を防ぎ、更に、可逆圧縮方式は限界値によって圧縮データを制限し、不可逆圧縮方式は出力圧縮データを多段階に分割して複数のセグメントに区切られたメモリに格納することにより、圧縮データ量を制御することができる。

〈第4実施例の変形〉上述の可逆圧縮部47は、可逆な圧縮方式に従って圧縮データをカウントする手段と、限界値Lを設定する手段と、それらカウント結果Dと限界値Lとを比較し、その結果を出力できる手段とを備えたものであれば、種々の変形が可能である。

【0066】上記第4実施例では、多段階に分割された圧縮データを複数のセグメントに区切られた圧縮メモリ7に格納する際に、段階数の大きいもの（段階4）から小さいもの（段階1）の順序で無効にして行ったが、本発明はこれに限定されるものではなく、適応的に選択して無効にして行っても良い。また、セグメント情報テーブルとして、図11に示すタイプを用いたが、これに限定されるものではなく、本発明の主旨を逸脱しない範囲であれば種々の変形が可能である。

【0067】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、上述の全ての実施例はその動作がハードウェアロジックによって実現されていたが、プログラムによって達成される場合にも適用できることは言うまで

21

【0068】もない。

【発明の効果】以上説明した様に、一時貯蔵手段において分離手段を不要としたことにより、システム全体の小型化とコスト減とが可能となった。

【図面の簡単な説明】

【図1】本実施例における画像処理装置の構成を示す概略ブロック図である。

【図2】ジグザグスキャン動作におけるスキャン方向を示す図である。

【図3】第1の実施例での適応ハフマン符号化部の構成¹⁰を示すブロック図である。

【図4】第1の実施例でのバッファメモリ15に格納されるデータを示す図である。

【図5】しきい値テーブルを示す図である。

【図6】セグメント毎に区切られた圧縮メモリの構成を示す図である。

【図7】セグメント情報テーブルの構成を示す図である。

【図8】圧縮データの格納処理を示すフローチャートである。²⁰

【図9】第2の実施例での適応ハフマン符号化部の構成を示すブロック図である。

【図10】第2の実施例でのバッファメモリ22に格納されるデータを示す図である。

【図11】第3の実施例での適応ハフマン符号化部の構成を示すブロック図である。

【図12】インデックステーブルを示す図である。

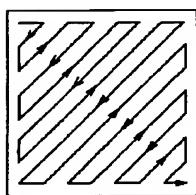
【図13】第4実施例での画像圧縮装置の構成を示すブロック図である。

【図14】第4の実施例での画像伸長装置の構成を示す³⁰ブロック図である。

【符号の説明】

- * 1 色変換部
- 2 サブサンプリング部
- 3 D C T 部
- 4 量子化部
- 5 適応ハフマン符号化部
- 6 セグメントコントローラ
- 7 圧縮メモリ
- 8 量子化テーブル
- 9 ハフマンテーブル
- 10 セグメント情報テーブル
- 11 C P U
- 12 R O M
- 13 R A M
- 14 ハフマン符号化部
- 15 バッファメモリ
- 16 基準値算出部
- 17 符号長加算器
- 18 比較器
- 19 しきい値テーブル
- 20 段階決定部
- 21 ハフマン符号化部
- 22 バッファメモリ
- 23 カウンタ
- 24 段階決定部
- 25 符号長加算器
- 26 比較器
- 27 しきい値テーブル
- 31 ハフマン符号化部
- 32 カウンタ
- 33 比較器
- 34 カウンタ
- * 35 インデックステーブル

【図2】



【図4】

DC成分 → AC成分 → DC成分 → AC成分 →

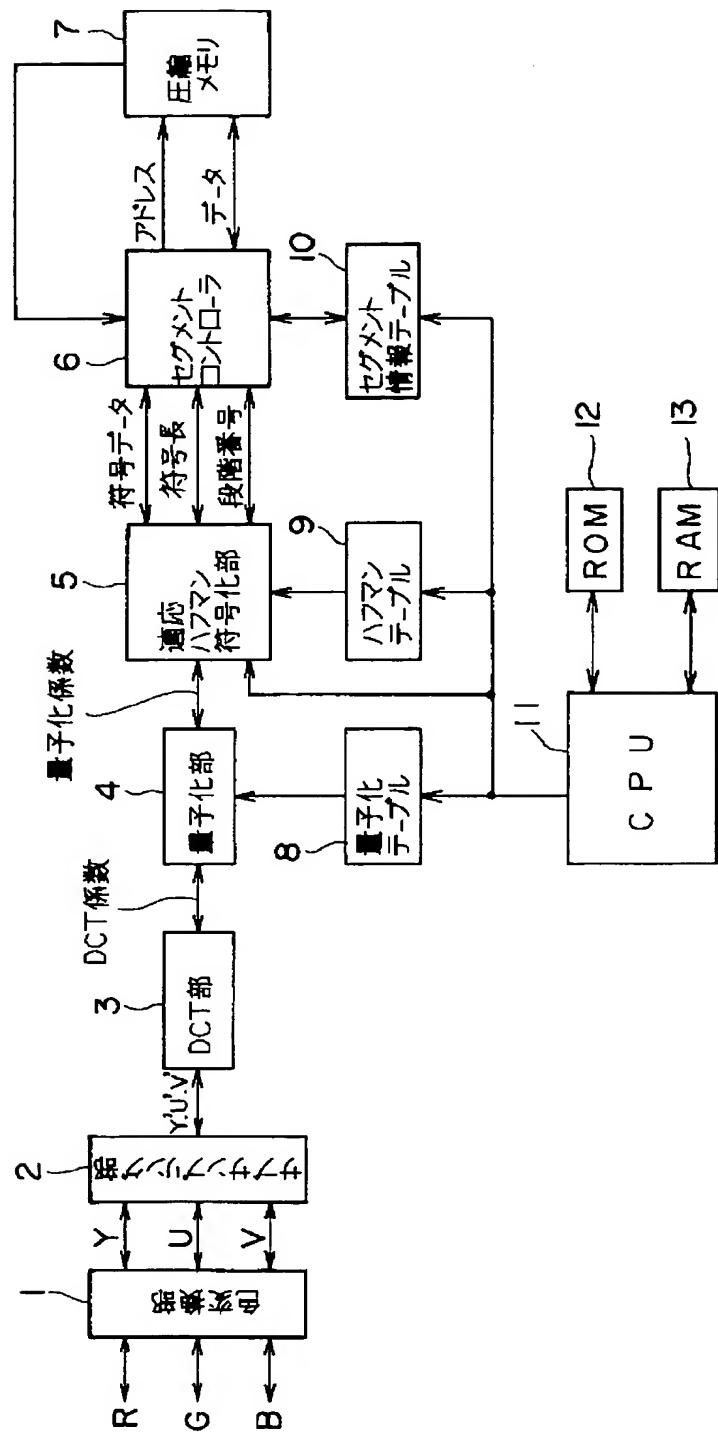
インデックス	符号データ	符号長	累積値
0	0x00001FF2	13	—
1	0x0003FFA5	18	18
2	0x00000007C	7	25
3	0x0000035B3	14	39
4	0x001FC59E	21	21
5	0x0000001AC	9	30
6	0x000072EC	15	45
.	.	.	.
.	.	.	.
.	.	.	.
18	0x00000001A	5	95
0	0x000037E2	14	—
1	0x003FA201	21	21
.	.	.	.

第1段階のデータ
累積値のリセット

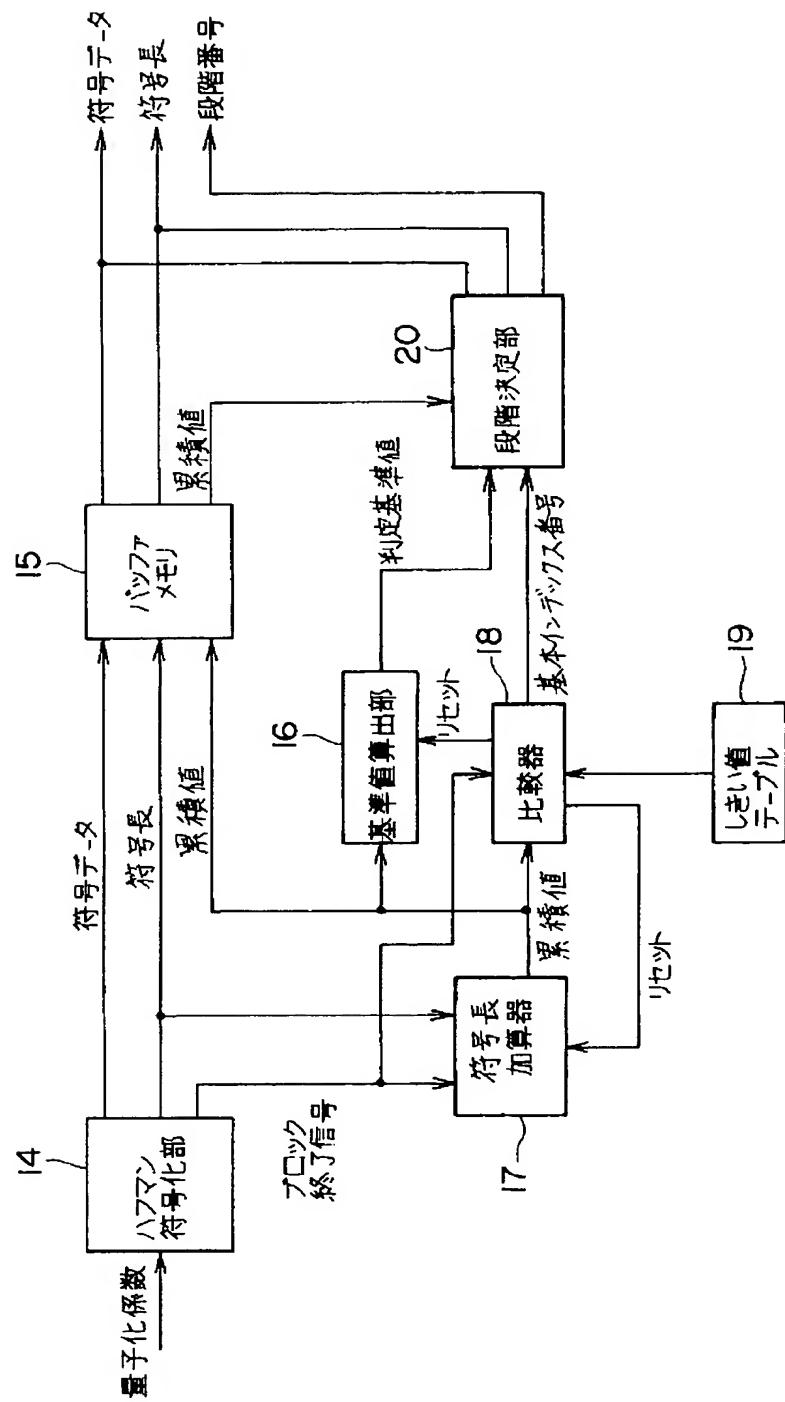
【図5】

色成分	閾値(ビット)
Y成分	40
U成分	28
V成分	28

【図1】



【図3】



【図6】

圧縮ビット	
セグメントS-1 (100KB)	
セグメントS-2 (100KB)	
セグメントS-3 (100KB)	
セグメントS-4 (100KB)	
⋮	
セグメントS-N (100KB)	

【図7】

セグメント情報テーブル											
(画像の先頭) → (画像の終端)											
No.	1	2	3	4	5	6	7	8	9	10	11
ステージ (1)	1	S-1	S-5	S-8	S-10	S-14	·	S-(N-1)	S-4	S-15	END
ステージ (2)	1	S-2	S-6	S-9	S-13	·	·	S-N	S-II	END	END
ステージ (3)	1	S-3	S-7	S-12	·	·	S-(N-2)	END	END	END	END
ステージ (4)	0	S-4	S-11	S-15	·	·	·	·	·	·	·

↓
I : ステージ有効
O : ステージ無効

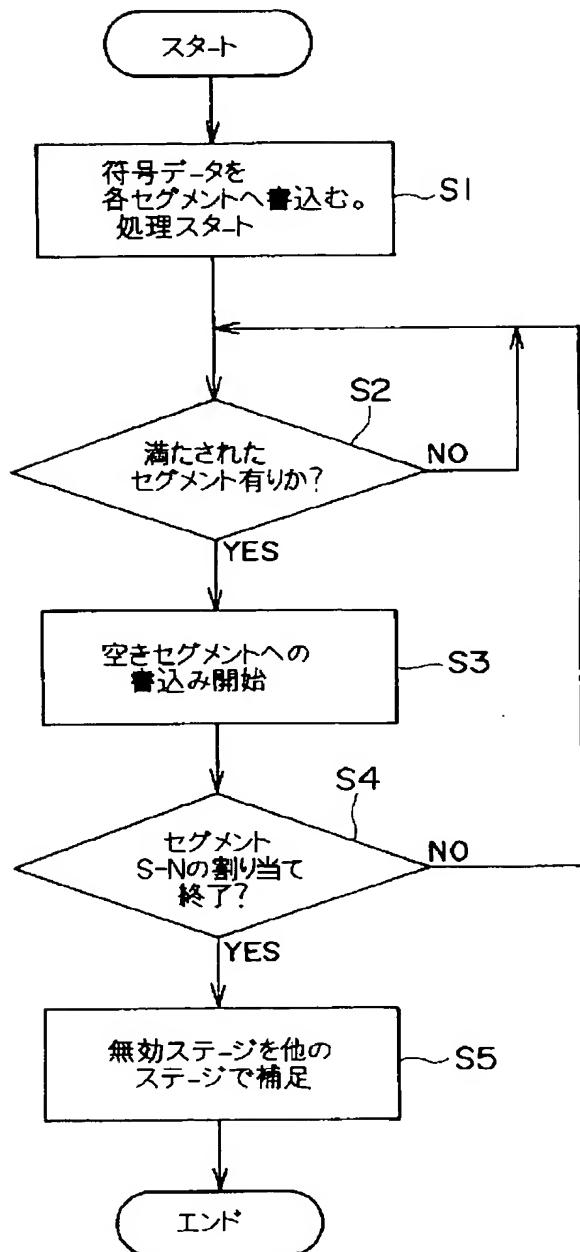
【図12】

【図10】

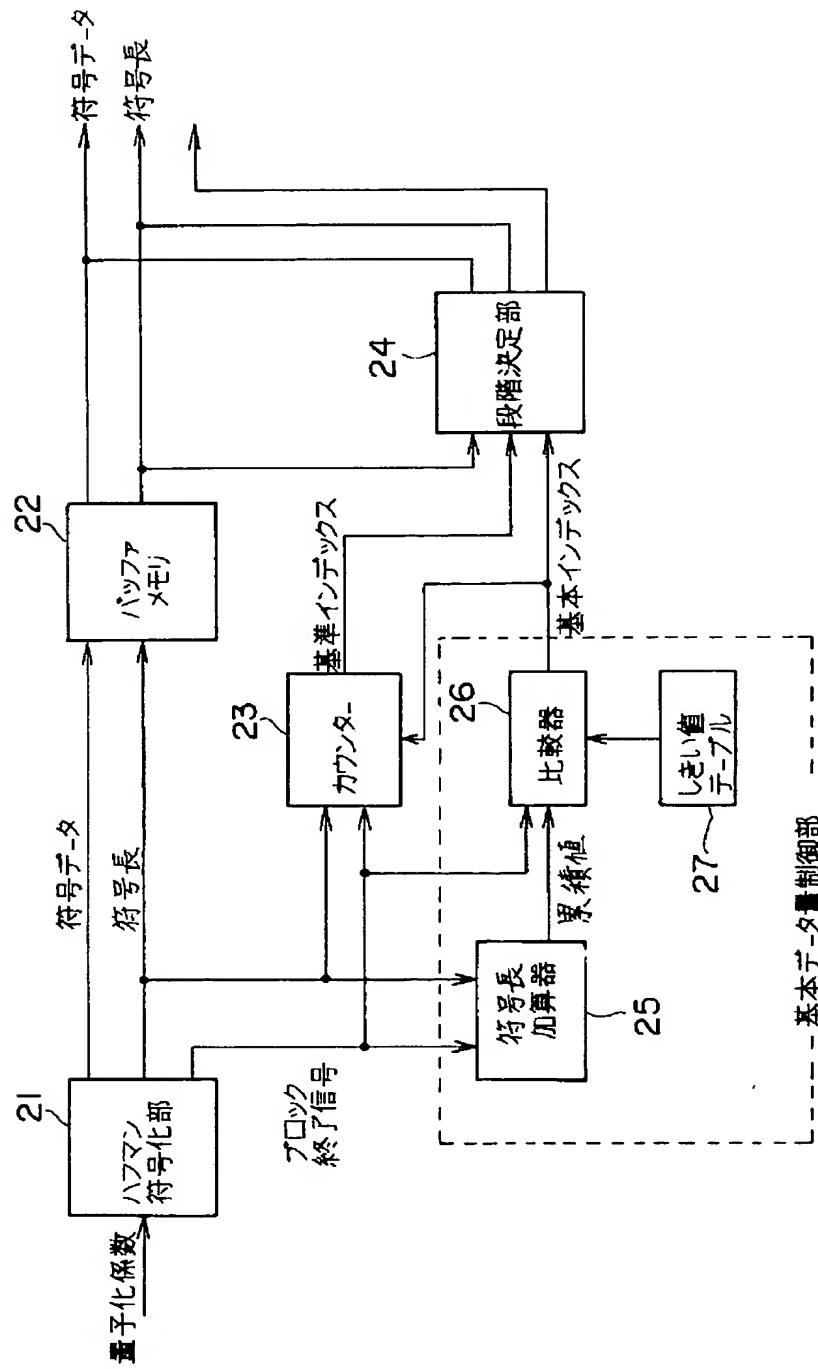
インデックス	符号データ	符号長
0	0x00001FF2	13
1	0x0003FFA5	18
2	0x0000007C	7
3	0x000035B3	14
4	0x001FC59E	21
5	0x000001AC	9
6	0x0000072EC	15
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
18	0x0000001A	5
0	0x000037E2	14
1	0x003FA201	21
⋮	⋮	⋮

	第1段階	第2段階	第3段階
Y成分	3	8	15
U成分	2	7	13
V成分	2	7	13

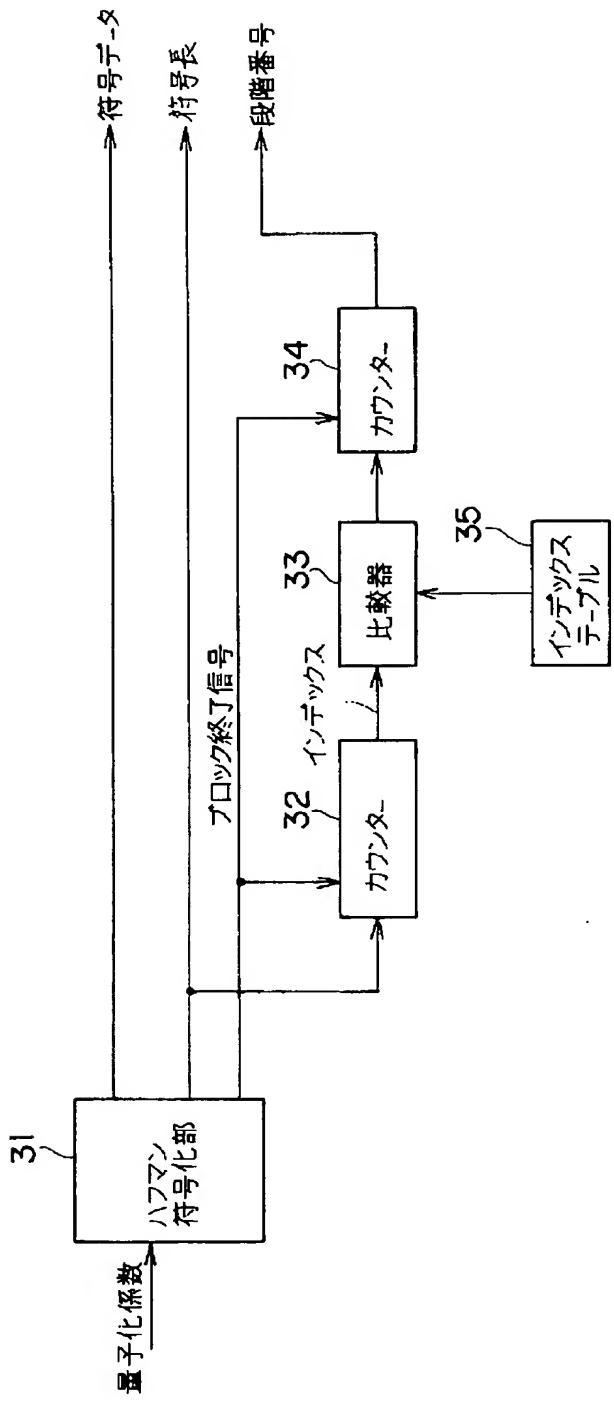
【図8】



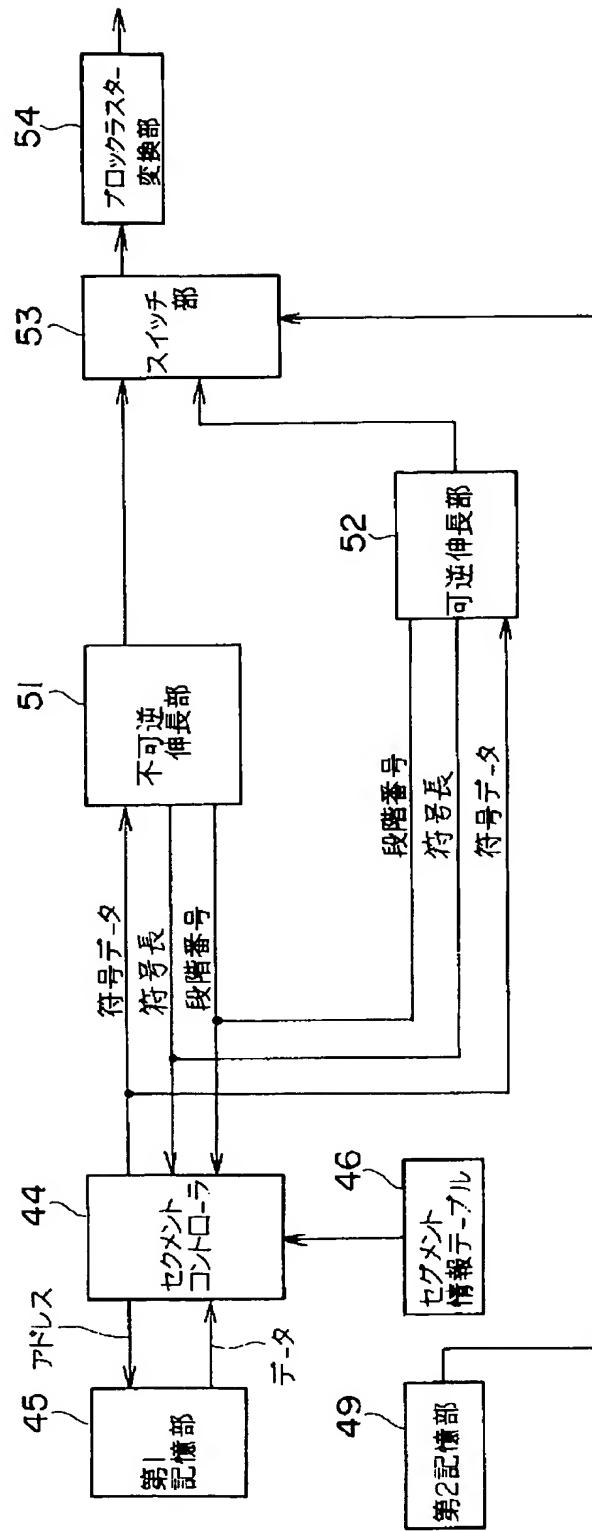
【図9】



【图 11】



【図14】



【図13】

